Abarna_day85

```
#include <stdio.h>
#include <string.h>
#define MOD 1000000007

int count_paths(int current_node, int pos, int L, char* S, char* node_chars, int** graph, int N, int*
visited) {
    if (pos == L) return 1;
    if (node_chars[current_node - 1] != S[pos]) return 0;

    int total_paths = 0;
    visited[current_node]++;

    for (int i = 1; i <= N; i++) {
        if (graph[current_node][i] && visited[i] < L) {
            total_paths = (total_paths + count_paths(i, pos + 1, L, S, node_chars, graph, N, visited))
% MOD;
        }
    }

    visited[current_node]--;
    return total_paths;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N, M, L;
        scanf("%d %d %d", &N, &M, &L);

        char S[1001], node_chars[101];
        scanf("%s", S);
        scanf("%s", node_chars);

        int u[1000], v[1000];
        for (int i = 0; i < M; i++) scanf("%d", &u[i]);
        for (int i = 0; i < M; i++) scanf("%d", &v[i]);

        int** graph = (int**)malloc((N + 1) * sizeof(int*));
        for (int i = 0; i <= N; i++) {
            graph[i] = (int*)calloc(N + 1, sizeof(int));
        }

        for (int i = 0; i < M; i++) {
            graph[u[i]][v[i]]++;
            graph[v[i]][u[i]]++;
```

```
        }

        int visited[101] = {0};
        int result = 0;

        for (int i = 1; i <= N; i++) {
            if (node_chars[i - 1] == S[0]) {
                result = (result + count_paths(i, 1, L, S, node_chars, graph, N, visited)) % MOD;
            }
        }

        printf("%d\n", result);

        for (int i = 0; i <= N; i++) {
            free(graph[i]);
        }
        free(graph);
    }
    return 0;
}
```